**Traffic Prediction using Image Generation and Machine Learning Models**

**1. Introduction**

The task involved predicting traffic conditions between customer and restaurant locations using a combination of **image-based deep learning** and **tabular machine learning models**. We explored two main approaches:

1. **Image-based classification with Convolutional Neural Networks (CNNs)**.
2. **Tabular classification with Logistic Regression and Random Forest**.

The dataset contained **customer and restaurant coordinates**, traffic condition labels (Low, Medium, High), and other attributes.

**2. Image Generation Process**

To apply CNNs, raw tabular data (latitude, longitude pairs) was transformed into **map images**.

* **Coordinate Extraction:** Customer and restaurant coordinates were parsed into numeric latitude and longitude.
* **Image Rendering:** Using StaticMap, a map tile was rendered with a **blue marker** for the customer location & a **red marker** for the restaurant location.A **line** connecting the two points.
* **Traffic Label Overlay:** The traffic condition (Low, Medium, High) was displayed at the top of the image.
* **Dataset Construction:** Each generated image was saved with its corresponding label (target\_bin).

This pipeline produced a dataset of labeled images suitable for CNN training.

**3. CNN Model Architecture**

The CNN was designed to classify the images into **three traffic categories**: Low, Medium, High.

**Model Structure:**

1. **Conv2D Layer (32 filters, 3×3, ReLU)**:-Detects low-level spatial features such as edges and points.
2. **MaxPooling2D (2×2)**:-Reduces spatial resolution to retain essential features while reducing computation.
3. **Conv2D Layer (64 filters, 3×3, ReLU)**:-Captures more complex patterns (routes, positional relationships between customer and restaurant).
4. **MaxPooling2D (2×2)**:-Further dimensionality reduction.
5. **Flatten Layer**:-Converts 2D feature maps into a 1D vector for dense layers.
6. **Dense Layer (128 units, ReLU)**:-Fully connected layer for learning global patterns.
7. **Dropout (0.3)**:-Regularization to prevent overfitting.
8. **Output Layer (Dense, 3 units, Softmax)**:-Multiclass classification into Low, Medium, or High traffic.

**Training Details:**

* Optimizer: Adam (lr=1e-3)
* Loss Function: Categorical Crossentropy (multiclass)
* Early Stopping: To prevent overfitting, with patience of 5 epochs

**4. Model Performance**

**Confusion Matrix for CNN**

[[ 0 0 5] → Class 'Low' misclassified as 'High'  
 [ 0 0 9] → Class 'Medium' misclassified as 'High'  
 [ 0 0 26]] → Class 'High' predicted correctly

This shows that the CNN was biased toward predicting the **High traffic** class, failing to distinguish between Low and Medium.

**Performance Metrics**

| **Model** | **Accuracy** | **Precision** | **Recall** | **F1** | **AUC** |
| --- | --- | --- | --- | --- | --- |
| CNN (images) | **0.650** | 0.423 | **0.650** | 0.512 | 0.516 |
| Logistic Regression (tabular) | 0.615 | 0.420 | 0.615 | 0.494 | – |
| Random Forest (tabular) | 0.570 | **0.466** | 0.570 | **0.504** | – |

**5. Comparison and Analysis**

* **CNN** achieved the **highest accuracy and recall**, but precision was low, indicating overprediction of the dominant class (High).
* **Logistic Regression** was competitive, with slightly lower accuracy than CNN.
* **Random Forest** had the best precision but overall weaker performance compared to CNN.

**6. Summary**

In this study, we explored traffic prediction using both image-based deep learning and tabular machine learning approaches. Tabular baselines included **Logistic Regression (Acc: 0.615, F1: 0.494)** and **Random Forest** **(Acc: 0.570, F1: 0.504)**, while a **CNN** trained on map images achieved the **best accuracy (0.650) and recall (0.650)** but struggled with **precision (0.423)**, reflecting its tendency to overpredict the "High" traffic class. These results show that while CNNs can capture spatial relationships between customer and restaurant locations more effectively than traditional models, their performance is limited by class imbalance. To improve predictive power, actionable steps include applying data augmentation to balance categories, experimenting with pretrained CNNs (e.g., ResNet, VGG) for better feature extraction, and exploring hybrid models that combine spatial image features with tabular attributes.

Overall, CNNs present a promising direction for traffic prediction, but careful handling of imbalance and model generalisation is key for more reliable deployment.

**Key Insight:**  
CNNs can learn from **spatially meaningful images** derived from tabular coordinates; however, care must be taken to handle class imbalance and improve generalisation.